ASUS Laptop Price Prediction

REDOAN-UR-RAHMAN (14.02.04.043)
ANIK CHOWDHURY (14.02.04.048)

Last Update - February 14, 2018

Contents

I Documentation of the Dataset 4

1 Dataset 4
- 1.1 Title 4
- 1.2 Sources 4
- 1.3 Number of Instances 4
- 1.4 Number of Attributes 4
- 1.5 Attribute Information 4
- 1.6 Missing Attribute Values 4
- 1.7 Attribute's Data Types 5
- 1.8 Summary Statistics 5
- 1.9 Correlation Of the Features 5

2 A brief description of the dataset 6
- 2.1 Data Dictionary 6

II Problem 7

3 A brief description of the problem 7

III Solution & Evaluation of the Solution 8

4 Models 8
- 4.1 Linear Regression 8
- 4.2 Support Vector Machines 8
- 4.3 Decision Tree Regressor 9
- 4.4 Extra Tree Regressor 9
- 4.5 Random Forest Regressor 10
- 4.6 Adaboost Regressor 10

5 Performance Metrics 11
- 5.1 Explained variance 11
- 5.2 Mean absolute error 11
- 5.3 Mean squared error 11
- 5.4 Mean squared logarithmic error 12
- 5.5 Median absolute error 12
- 5.6 R^2 score, the coefficient of determination 12

IV Conclusion 14

6 Discussion 14

V Readme 15

7 How to run **15**
7.1 Before Start : 15
7.2 After installation : 15

Part I
Documentation of the Dataset

1 Dataset

1.1 Title

ASUS Laptop Pricing Dataset is updated February 09, 2018 by REDOAN-UR-RAHMAN & ANIK CHOWDHURY

1.2 Sources

Datas are collected from these websites.

1. Computer Source Bangladesh [6]
2. Daffodil Computers Ltd [7]
3. Global Band Limited [10]
4. Ryans Computers [16]

1.3 Number of Instances

The dataset contains a total of 100 cases.

1.4 Number of Attributes

There are 9 attributes in each case of the dataset.

1.5 Attribute Information

- **cores-** Number of cores in the processor of the laptop.
- **threads-** Number of threads in the processor of the laptop.
- **speed-** Clock speed of the processor in MHz.
- **ram-** Random Access Memory size in GB.
- **ddr-** Double data rate type of the RAM.
- **gpu-** Memory size of Graphics Processing Unit used by the laptop in MB.
- **hdd-** Hard Disk Drive Size in GB.
- **ssd-** Solid State Drive Size in GB.
- **cost-** Price of the Laptop.(Target Attribute)

1.6 Missing Attribute Values

None.

1.7 Attribute's Data Types

Table 1: Attribute with their data type

Attribute Name	Data Type
cores	**int64**
threads	**int64**
speed	**int64**
ram	**int64**
ddr	**int64**
gpu	**int64**
hdd	**int64**
ssd	**int64**
cost	**int64**

1.8 Summary Statistics

Table 2: DataSet Summary Statistics

	cores	threads	speed	ram	ddr	gpu	hdd	ssd	cost
count	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
mean	2.6	4.8	2504.5	10.3	3.8	2288.0	835.1	190.8	80439.0
std	0.9	1.9	562.3	10.9	0.4	2207.2	546.1	271.4	59964.4
min	2.0	2.0	1100.0	2.0	3.0	0.0	0.0	0.0	21600.0
25%	2.0	4.0	2300.0	4.0	4.0	0.0	500.0	0.0	43000.0
50%	2.0	4.0	2500.0	8.0	4.0	2000.0	1000.0	0.0	69750.0
75%	4.0	4.0	2825.0	8.0	4.0	4000.0	1000.0	256.0	91000.0
max	4.0	8.0	3800.0	64.0	4.0	8000.0	2000.0	1000.0	398500.0

1.9 Correlation Of the Features

Table 3: Correlation Matrix

	cores	threads	speed	ram	ddr	gpu	hdd	ssd
cores	1.00	0.84	0.23	0.51	0.23	0.76	0.32	0.19
threads	0.84	1.00	0.42	0.59	0.41	0.77	0.31	0.32
speed	0.23	0.42	1.00	0.47	0.63	0.55	-0.11	0.59
ram	0.51	0.59	0.47	1.00	0.22	0.73	-0.11	0.65
ddr	0.23	0.41	0.63	0.22	1.00	0.42	0.12	0.26
gpu	0.76	0.77	0.55	0.73	0.42	1.00	0.09	0.49
hdd	0.32	0.31	-0.11	-0.11	0.12	0.09	1.00	-0.59
ssd	0.19	0.32	0.59	0.65	0.26	0.49	-0.59	1.00

2 A brief description of the dataset

The data-set consists of 9 columns among which 8 are features and the last column is the target class. The features are different hardware specifications of a Laptop. Depending on the specifications we tried to predict their prices.

2.1 Data Dictionary

Table 4: Data Dictionary

Features	**Definition**
cores	Number of cores in the processor of the laptop.
threads	Number of threads in the processor of the laptop
speed	Clock speed of the processor in MHz
ram	Random Access Memory size in GB
ddr	Double data rate type of the RAM
gpu	Memory size of Graphics Processing Unit used by the laptop in MB
hdd	Hard Disk Drive Size in GB
ssd	Solid State Drive Size in GB
cost	Price of the Laptop

Part II
Problem

3 A brief description of the problem

New computing technologies has made machine learning today different from the machine learning of the past. It was born from pattern recognition and the theory that computers can learn without being programmed to perform specific tasks; researchers interested in artificial intelligence wanted to see if computers could learn from data. The iterative aspect of machine learning is important because as models are exposed to new data, they are able to independently adapt. They learn from previous computations to produce reliable, repeatable decisions and results. Its a science thats not new but one that has gained fresh momentum.
In this project we have tried to make a data set and solve a problem. As a computer engineer in Bangladesh we are to face some common questions about the pricing of the electronic device. So our idea was to make the computer predict the price of Laptop which are build by a popular brand in Bangladesh **ASUS [4]** based on their Hardware specification.

Part III

Solution & Evaluation of the Solution

4 Models

We have used 6 models in this problem to see which one gives the better accuracy. Score for each model is tested individually. The data-set is first standardized by using scalar transform so that no feature has more priority over the other. Then it is split for training and testing. 80% of data is used for training and the rest 20% is used for testing purpose.

4.1 Linear Regression

Linear regression is a linear approach for modelling the relationship between a scalar dependent variable y and one or more explanatory variables (or independent variables) denoted X. The case of one explanatory variable is called simple linear regression. For more than one explanatory variable, the process is called multiple linear regression. In linear regression, the relationships are modeled using linear predictor functions whose unknown model parameters are estimated from the data. Such models are called linear models. [11]
Linear Regression in sklearn:
class sklearn.linear_model.LinearRegression(fit_intercept=True, normalize=False,-copy_X=True, n_jobs=1)
where,

fit_intercept : boolean, optional, default True
Whether to calculate the intercept for this model. If set to False, no intercept will be used in calculations (e.g. data is expected to be already centered).

normalize : boolean, optional, default False
This parameter is ignored when fit_intercept is set to False. If True, the regressors X will be normalized before regression by subtracting the mean and dividing by the $l_2 norm$. If you wish to standardize, please use sklearn.preprocessing.StandardScaler before calling fit on an estimator with normalize=False.

copy_X : boolean, optional, default True
If True, X will be copied; else, it may be overwritten.

n_jobs : int, optional, default 1
The number of jobs to use for the computation. If -1 all CPUs are used. This will only provide speedup for n_targets > 1 and sufficient large problems.

[18]

4.2 Support Vector Machines

In machine learning, support vector machines (SVMs, also support vector networks) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier (although methods such as Platt scaling exist to use SVM in a probabilistic classification setting). An SVM

model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall. [21]
The implementation is based on libsvm. The fit time complexity is more than quadratic with the number of samples which makes it hard to scale to data-set with more than a couple of 10000 samples.
We use ***svm.SVR(kernel=rbf)*** where,

kernel : string, optional (default =rbf)
Specifies the kernel type to be used in the algorithm. It must be one of linear, poly, rbf, sigmoid, precomputed or a callable. If none is given, rbf will be used. We have used linear, poly and rbf kernel to compute the accuracy of our dataset.

[19]

4.3 Decision Tree Regressor

Decision tree builds regression or classification models in the form of a tree structure. It brakes down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes. A decision node has two or more branches, each representing values for the attribute tested. Leaf node represents a decision on the numerical target. The topmost decision node in a tree which corresponds to the best predictor called root node. Decision trees can handle both categorical and numerical data. [8] We use ***DecisionTreeRegressor(random_state=33)*** where,

random_state : int, RandomState instance or None, optional (default=None)
If int, random_state is the seed used by the random number generator; If RandomState instance, random_state is the random number generator; If None, the random number generator is the RandomState instance used by np.random.

[20]

4.4 Extra Tree Regressor

Adding one further step of randomization yields extremely randomized trees, or ExtraTrees. These are trained using bagging and the random subspace method, like in an ordinary random forest, but additionally the top-down splitting in the tree learner is randomized. Instead of computing the locally optimal feature/split combination, for each feature under consideration, a random value is selected for the split. This value is selected from the feature's empirical range. [1]
We use ***ensemble.ExtraTreesRegressor(n_estimators=10, random_state=42)*** where,

n_estimators : int, integer, optional (default=10)
The number of trees in the forest.

random_state : int,RandomState instance or None, optional (default=None)
If int, random_state is the seed used by the random number generator; If RandomState instance, random_state is the random number generator; If None, the random number generator is the RandomState instance used by np.random.

[3]

4.5 Random Forest Regressor

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set. [15] We use ***RandomForestRegressor(n_estimators=10, max_depth=None, random_state=33)*** where,

n_estimators : integer, optional (default=10)
The number of trees in the forest.

max_depth : integer or None, optional (default=None)
The maximum depth of the tree. If None, then nodes are expanded until all leaves are pure or until all leaves contain less than min_samples_split samples.

random_state : int, RandomState instance or None, optional (default=None)
If int, random_state is the seed used by the random number generator; If RandomState instance, random_state is the random number generator; If None, the random number generator is the RandomState instance used by np.random.

[2]

4.6 Adaboost Regressor

An AdaBoost regressor is a meta-estimator that begins by fitting a regressor on the original dataset and then fits additional copies of the regressor on the same dataset but where the weights of instances are adjusted according to the error of the current prediction. As such, subsequent regressors focus more on difficult cases. [17] We use ***AdaBoostRegressor()***

5 Performance Metrics

For testing the performance we have chosen 6 performance matrices which are -

- Explained variance
- Mean absolute error
- Mean squared error
- Mean squared logarithmic error
- median absolute error
- R^2 score

Some sort description about these errors are given below:

5.1 Explained variance

In statistics, explained variation measures the proportion to which a mathematical model accounts for the variation (dispersion) of a given data set. Often, variation is quantified as variance; then, the more specific term explained variance can be used. [9]
If ŷ is the estimated target output, y the corresponding (correct) target output, and Var is Variance, the square of the standard deviation, then the explained variance is estimated as follow:

$$explained\_variance(y, \hat{y}) = 1 - \frac{Var\{y, \hat{y}\}}{Var\{y\}}$$

The best possible score is 1.0, lower values are worse. [14]

5.2 Mean absolute error

In statistics, mean absolute error (MAE) is a measure of difference between two continuous variables. [12]
The mean_absolute_error function computes mean absolute error, a risk metric corresponding to the expected value of the absolute error loss or l_1-norm loss.
If $\hat{y}_i$ is the predicted value of the i^{th} sample, and y_i is the corresponding true value, then the mean absolute error (MAE) estimated over n_{samples} is defined as

$$MAE(y, \hat{y}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} |y_i - \hat{y}_i|$$

[14]

5.3 Mean squared error

In statistics, the mean squared error (MSE) or mean squared deviation (MSD) of an estimator (of a procedure for estimating an unobserved quantity) measures the average of the squares of the errors or deviationsthat is, the difference between the estimator and what is estimated. [13]

The mean_squared_error function computes mean square error, a risk metric corresponding to the expected value of the squared (quadratic) error or loss.
If $\hat{y}_i$ is the predicted value of the i^{th} sample, and y_i is the corresponding true value, then the mean squared error (MSE) estimated over n_{samples} is defined as

$$MSE(y, \hat{y}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} (y_i - \hat{y}_i)^2$$

[14]

5.4 Mean squared logarithmic error

The mean_squared_log_error function computes a risk metric corresponding to the expected value of the squared logarithmic (quadratic) error or loss.
If $\hat{y}_i$ is the predicted value of the i^th sample, and y_i is the corresponding true value, then the mean squared logarithmic error (MSLE) estimated over n_{samples} is defined as

$$MSLE(y, \hat{y}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} (\log_e(1 + y_i) - \log_e(1 + \hat{y}_i))^2$$

[14]
Where $\log_e(x)$ means the natural logarithm of x. This metric is best to use when targets having exponential growth, such as population counts, average sales of a commodity over a span of years etc.
Note that this metric penalizes an under-predicted estimate greater than an over-predicted estimate. [14]

5.5 Median absolute error

The median_absolute_error is particularly interesting because it is robust to outliers. The loss is calculated by taking the median of all absolute differences between the target and the prediction.
If $\hat{y}_i$ is the predicted value of the i^th sample and y_i is the corresponding true value, then the median absolute error (MedAE) estimated over n_{samples} is defined as

$$MedAE(y, \hat{y}) = median(|y_1 - \hat{y}_1|, ..., |y_n - \hat{y}_n|)$$

[14]

5.6 R^2 score, the coefficient of determination

In statistics, the coefficient of determination, denoted R2 or r2 and pronounced "R squared", is the proportion of the variance in the dependent variable that is predictable from the independent variable(s). [5]
The r2_score function computes R^2, the coefficient of determination. It provides a measure of how well future samples are likely to be predicted by the model. Best possible score is 1.0 and it can be negative (because the model can be arbitrarily worse). ***A constant model that always***

predicts the expected value of y, disregarding the input features, would get a R^2 score of 0.0.

If $\hat{y}_i$ is the predicted value of the $i_t h$ sample and y_i is the corresponding true value, then the score R_2 estimated over n_{samples} is defined as

$$R_2(y, \hat{y}) = 1 - \frac{\sum_{i=0}^{n_{samples}-1} (y_i - \hat{y}_i)^2}{\sum_{i=0}^{n_{samples}-1} (y_i - \bar{y}_i)^2}$$

where

$$\bar{y} = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} y_i$$

[14]

The comparison of the performance scores for each model is shown below:

Table 5: Performance Scores

	Linear	**SVM Linear**	**SVM Poly**	**SVM rbf**	**Decision Tree**	**Extra Tree**	**Random Forest**	**Adaboost**
Explained Variance	0.9489	0.0069	0.0055	0.0001	0.9490	0.9583	0.9728	0.9292
Mean Absolute Error	8440.2539	29629.7587	29708.1254	29748.9878	8159.1667	7128.6667	6759.8750	10353.1187
Mean Squared Error	177024739.9438	3337461575.4129	3341471058.9961	3359758454.8314	173556680.5556	150104824.4444	91636193.0903	256880683.4650
Mean Squared Log Error	0.0158	0.3111	0.3126	0.3143	0.0165	0.0099	0.0129	0.0310
Median Absolute Error	3385.7773	17519.9203	17501.0728	17505.8802	4000.0000	3500.0000	4075.0000	6233.6601
R Square Score	0.9471	0.0023	0.0011	-0.0044	0.9481	0.9551	0.9726	0.9232

Part IV
Conclusion

6 Discussion

Amongst the models we have used to solve our targeted problem RandomForest Regressor gives the best result. Because RandomForest Regressor works with randomly selected features and fit them in the model which results better prediction.

Part V
Readme

7 How to run

7.1 Before Start :

If you want to run our project and evaluate, what we have done. You have to install the following softwars-

- Python 3.6.3 [How to install]
- Anaconda Navigator 1.6.11 [How to install]

7.2 After installation :

After completing installation of ***Python 3.6.3*** and ***Anaconda Navigator 1.6.11*** you need to take the following steps:

1. Run ***Anaconda Navigator***
2. Run ***Spyder***
3. Open the file ***AsusLaptopPricing.py*** using ***Spyder***.
4. Run the code.

N.B.Keep "**AsusLaptopPricing.py**" & "**Price.csv**" both files in a same folder.

References

[1] 1.11. ensemble methods. [Online; accessed 14-February-2018].

[2] 3.2.4.3.2. sklearn.ensemble.randomforestregressor. [Online; accessed 14-February-2018].

[3] 3.2.4.3.4. sklearn.ensemble.extratreesregressor. [Online; accessed 14-February-2018].

[4] Asus bangladesh. [Online; accessed 14-February-2018].

[5] Coefficient of determination. [Online; accessed 14-February-2018].

[6] Computer source ltd (csl) is the largest technology distributor of bangladesh. [Online; accessed 09-February-2018].

[7] Daffodil computers ltd. [Online; accessed 09-February-2018].

[8] Decision tree. [Online; accessed 14-February-2018].

[9] Explained variation. [Online; accessed 14-February-2018].

[10] Global band pvt. ltd. [Online; accessed 09-February-2018].

[11] Linear regression. [Online; accessed 14-February-2018].

[12] Mean absolute error. [Online; accessed 14-February-2018].

[13] Mean squared error. [Online; accessed 14-February-2018].

[14] Model evaluation: quantifying the quality of predictions. [Online; accessed 14-February-2018].

[15] Random forest. [Online; accessed 14-February-2018].

[16] Ryans computers - largest retail chain stores for computer product in bangladesh. [Online; accessed 09-February-2018].

[17] sklearn.ensemble.adaboostregressor. [Online; accessed 14-February-2018].

[18] sklearn.linear$_{m}odel.linearregression.[Online;accessed14 - February - 2018$].

[19] sklearn.svm.svr. [Online; accessed 14-February-2018].

[20] sklearn.tree.decisiontreeregressor. [Online; accessed 14-February-2018].

[21] Support vector machine. [Online; accessed 14-February-2018].